Eurographics Workshop on Urban Data Modelling and Visualisation (2013)
V. Tourre and G. Besuievsky (Editors)

Interactive techniques for populating large virtual cities

K. Jordao[1] J. Pettré[1] M.P. Cani[2]

[1]Inria Rennes, France
[2]Laboratory Jean Kuntzmann / Inria Grenoble, France

Abstract

For many applications, exploring empty virtual cities is not sufficient: streets and squares need to be populated by crowds of virtual humans. This position paper addresses the problem of generating such crowds when the scale of the city prevents the use of standard simulation methods. We first present the concept of crowd patches and review their advantages and drawbacks for animating large crowds of people. We then discuss the goals of our future work, namely being able to interactively design such large populations.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Virtual cities are now easily available to the general public with various purposes, ranging from digital tourism to architectural decision making. However, virtual cities usually remains empty of any people. Populating such virtual cities would make them more lively, convincing, and would enhance level of realism and of credibility. We address the question of populating interactive and large virtual environments. Our goal is to provide efficient techniques for both designing virtual populations and displaying animated cities to the users.

Crowd simulation is a natural solution to the problem of populating interactive environments [TM07]. However, such techniques do not answer some specific needs in the task of populating large scale environments. Firstly, simulations of virtual crowds are computationally expensive in terms of both time and memory usage. Hence, the size limitation of the virtual environments. Secondly, it is difficult to match a simulation setup with a desired content because of the chaotic nature of crowd simulation.

The ANR-CHROME project gathers Inria and the ArchiVideo (http://www.archivideo.com/) and Golaem (http://www.golaem.com/) companies, to tackle the problem of populating large scale environments. As an illustrative objective, the project aims at populating a 3D map of France: Territoire3D (http://www.territoire3d.com/).

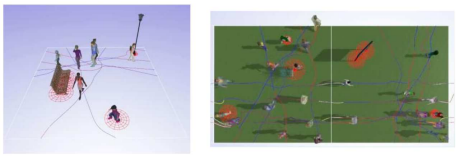

Figure 1: *Crowd patch content and connectivity.* ***Left****: A crowd patch with its obtacles and human trajectories.* ***Right****: Two patches plugged because of the similarity between their spatiotemporal conditions at their limits.*

2. Crowd Patches

We previously proposed approaches to efficiently populate virtual cities and to design digital populations [YMPT09, LCS*12]. We introduced the concept of crowd patches. The key idea of this method is to precompute some portions of crowd animations, as displayed in Figure 1. Inside a patch, animations are computed to be periodic over a constant period of time. This allows endlessly replaying precomputed trajectories with smooth animation. However, animated humans (more generally moving objects like cars, dogs, birds) may cross the limits of a patch and move to a neighbor patch. The inputs and outputs on the patch sides set spatiotemporal constraints. Playing on the constraints compatibility between different patches, we can interconnect them. Thus, we create large scale animated environments by assembling patches like a jigsaw puzzle.

Compared to crowd simulation based approches, the crowd patches method has some relevant properties in the task of populating virtual cities. Firstly this method is not expensive in computational resources. Once a patch is computed, animation is obtained from a simple data replay. Complexity is linear with the number of animated objects. Secondly the method is not memory consuming. A same patch can be reused at different places. With only few patches we can populate large enviroment. In summary we repeat patches contained in space and time to optimize computing and memory cost.

Animating populations based on crowd patches is reduced to animation replay with reuse of animations in both space and time. Especially, animations contained into patches have to: (i) be periodic and (ii) satisfy spatiotemporal constraints.

3. Two Methods to Compute Crowd Patches

In this section we describe two different approach to create crowd patches.

Figure 2: *Results from Yersin's method*

3.1. Simulation Method

In the first method, described in [YMPT09], we start from a set of constraints at the edges of patches. Then we deduce a set of trajectories which satisfy these constraints.

The key idea is the following: as in the social forces model [HM95], humans repulse each other to avoid collision. They are attracted toward their goal which is a point at the limit of the patch. In addition, they have to meet a spatiotemporal constraint, i.e. go through a waypoint at a given time: closer is this time, stronger the human is attracted. As a result, this method computes trajectories which respect the given constraints. Then, it is relatively easy to generate what we end up with a set of interconnectable patches (as shown in Figure 2). However some trajectories can result in unnatural behaviour.

3.2. Data Driven Method

In the second method, described in [LCS*12], instead of generating trajectories, we use existing data (motion capture or simulation data as shown in Figure 3). We search for portions of data which satisfy two rules. Firstly data have to be as periodic as possible. Secondly the spatiotemporal conditions in the data limit have to match with other patches

Figure 3: *A result from Li's method.* ***Left****: A snapshot of motion capture session, where pedestrians followed each other.* ***Right****: Periodicity is found in motion capture's data and then transformed into connectable patches.*

constraints in order to be assembled. Data portions are never perfectly periodic or never perfectly connected. As a result, additional deformation is applied to data to strictly satisfy these conditions. As a summary, this method enables copying and pasting existing crowd data into new environments.

4. Future Works and Conclusions

Crowd patches can be computed by either the Yersin or Li's method. The most needed tools now concerns authoring crowd in virtual environment. In a near future work, we want to provide designers with the ability to interactively design large populated environment. Designers will be able to use libraries of precomputed patches to compose new populations, by simply drag and drop them into their environments.

In addition, our goal is to enable designing populations by working patch as a *clay*. This idea is based on the edition of mutable structured shapes [MWCS13]. The idea is to design the global shape in which the crowd will be animated. Then we convert each part of the global shape into a crowd patch. The designers would deform patches to fit them into desired shape.

References

[HM95] HELBING D., MOLNÁR P.: Social force model for pedestrian dynamics. *Phys. Rev. E 51* (May 1995), 4282–4286. 2

[LCS*12] LI Y., CHRISTIE M., SIRET O., KULPA R., PETTRÉ J.: Cloning crowd motions. In *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation* (Lausanne, Switzerland, 2012), Lee J., Kry P., (Eds.), Eurographics Association, pp. 201–210. 1, 2

[MWCS13] MILLIEZ A., WAND M., CANI M.-P., SEIDEL H.-P.: Mutable elastic models for sculpting structured shapes. *Computer Graphics Forum* (2013). 3

[TM07] THALMANN D. T., MUSSE S. R.: *Crowd Simulation.* British Library, 2007. 1

[YMPT09] YERSIN B., MAÏM J., PETTRÉ J., THALMANN D.: Crowd patches: populating large-scale virtual environments for real-time applications. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2009), I3D '09, ACM, pp. 207–214. 1, 2